

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-349652

(P2000-349652A)

(43) 公開日 平成12年12月15日 (2000. 12. 15)

(51) Int. Cl. ⁷	識別記号	F I	テーマコード(参考)
H 0 3 M 13/00		H 0 3 M 13/00	5 J 0 6 5
// G 1 1 B 20/18	5 5 0	G 1 1 B 20/18	5 5 0 C
	5 7 2		5 7 2 Z

審査請求 未請求 請求項の数9 O L (全 25 頁)

(21) 出願番号 特願平11-159613

(22) 出願日 平成11年6月7日 (1999. 6. 7)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 塚田 稔

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者 中村 一男

東京都小平市上水本町五丁目20番1号 株

式会社日立製作所半導体グループ内

(74) 代理人 100087170

弁理士 富田 和子

最終頁に続く

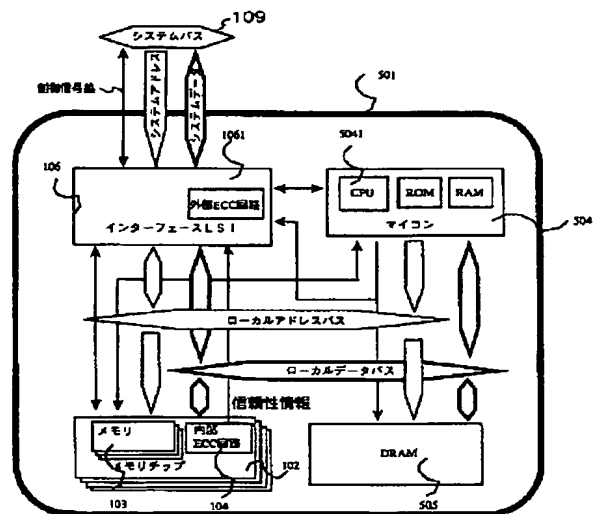
(54) 【発明の名称】 誤り訂正手段を備えた記憶装置

(57) 【要約】

【課題】 十分なデータ信頼率が得られるとともに、メモリチップの記憶素子のデータ信頼率の変化に柔軟に対応することができる記憶装置を提供する。

【解決手段】 メモリチップ102とインターフェースLSI 106とを備えたフラッシュメモリカード501において、メモリチップ102に内部ECC回路104を設け、インターフェースLSI 106に、外部ECC回路1061を設ける。そして、積符号による符号化を行う際は、まず、外部ECC回路1061によって外符号化を行い、更に、内部ECC回路104によって内符号化を行う。また、データ読み出し時は、内部ECC回路104によって内符号の復号を行い、行データの信頼性情報をメモリチップ102の外部に出力する。外部ECC回路1061は、外符号の復号を行う場合、行データの信頼性情報を利用して復号を行う。

図2



1

【特許請求の範囲】

【請求項1】 データを記憶する記憶手段と、
当該記憶手段に対するデータの読み書きを制御する制御手段とを備えた記憶装置であって、
前記記憶手段は、
データを格納するメモリ手段と、
当該メモリ手段に書き込むデータの誤り訂正符号化を行い、また、前記メモリ手段から読み出したデータの誤り検出及び誤り訂正を行う内部ECC手段とを備え、
当該内部ECC手段によって行った誤り訂正に関する信頼性情報を出力し、前記制御手段は、
前記記憶手段に書き込むデータの誤り訂正符号化を行い、また、前記記憶手段から読み出したデータの誤り検出及び誤り訂正を行う外部ECC手段を備え、
当該外部ECC手段は、前記記憶手段から出力された信頼性情報を利用して誤り訂正を行うことを特徴とする記憶装置。

【請求項2】 前記制御手段は、
ホストから書き込まれたデータを、外部ECC手段によって、外符号化し、
前記記憶手段は、
前記制御手段によって書き込まれた外符号を、内部ECC手段によって、内符号化し、前記メモリ手段に書き込むことを特徴とする請求項1に記載の記憶装置。

【請求項3】 第1の動作モードでは、前記外部ECC手段のみによる符号化を行い、第2の動作モードでは、前記内部ECC手段のみによる符号化を行い、第3の動作モードでは、前記外部ECC手段及び内部ECC手段による積符号化を行うことを特徴とする請求項1又は請求項2に記載の記憶装置。

【請求項4】 前記内部ECC手段及び外部ECC手段は、誤り訂正符号として、シンボル単位訂正符号で、組織符号となる巡回符号を用いることを特徴とする請求項1～3のいずれか一項に記載の記憶装置。

【請求項5】 データの記憶を行う記憶手段に対するデータの読み書きを制御するコントローラであって、
前記記憶手段に書き込むデータを誤り訂正符号化し、また、前記記憶手段から読み出したデータの誤りの有無を検出し、誤りが検出された場合に当該データの誤りを訂正する外部ECC手段を有し、
当該外部ECC手段は、
前記記憶手段から出力された、誤り訂正に関する信頼性情報を利用して、読み出しデータの誤り訂正を行うことを特徴とするコントローラ。

【請求項6】 前記外部ECC手段は、誤り訂正符号として、シンボル単位訂正符号で、組織符号となる巡回符号を用いることを特徴とする請求項5に記載のコントローラ。

【請求項7】 データを格納するメモリ手段と、
当該メモリ手段に書き込むデータの誤り訂正符号化を行

2

い、また、前記メモリ手段から読み出したデータの誤り検出及び誤り訂正処理を行う内部ECC手段とを備え、
当該内部ECC手段で行われた誤り訂正処理に関する信頼性情報を出力することを特徴とするメモリチップ。

【請求項8】 前記内部ECC回路は、誤り訂正能力を変更できることを特徴とする請求項7に記載のメモリチップ。

【請求項9】 前記内部ECC手段は、誤り訂正符号として、シンボル単位訂正符号で、組織符号となる巡回符号を用いることを特徴とする請求項7又は請求項8に記載のメモリチップ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、デジタル情報を記憶する記憶装置、特に、フラッシュメモリ等の半導体メモリを使った記憶装置に関する。

【0002】

【従来の技術】従来、コンピュータの主記憶装置には、RAM等の半導体メモリが記憶媒体として利用されてきたが、このような記憶媒体における誤り訂正方式は、例えば、特開平3-5995号公報に記載されているように、ビット単位の誤り訂正符号(Error Correcting Code: ECC)を用いて誤りを訂正していた。これは、半導体記憶媒体の1素子あたりの記録単位が1ビットで、素子のデータ誤り確率がランダムであることに起因している。この傾向は不揮発性メモリであるフラッシュメモリでも同様である。

【0003】一方、ハードディスクやCD-ROMなど、ランダム誤りとバースト誤りが混在する媒体においては、データをバースト誤り訂正ECCと、ランダム誤り訂正ECCとで二重に符号化し、データ復号時に制御信号により、それぞれの復号結果のうち1つを選択して出力する方法や、同じECCを用いて、二重に符号化し、それぞれのECCを連携させて、データを高信頼化させる積符号と呼ばれる方法が採用されている。前者の方法は、例えば、特開平2-301226号公報や特開平3-212026号公報に記載されている。また、後者の方法は、特開平7-202717号公報に記載されている。

【0004】以下、誤り訂正ECCと積符号について説明する。まず、誤り訂正ECCについて説明する。誤り訂正で用いられているECCにはいろいろあるが、まとまったバイトを1ブロックとして誤り訂正/検出を行うECCとして、リードソロモン符号(以下、RS符号という)がある。RS符号は、組織符号となる巡回符号である。組織符号とは、情報データ部と冗長データ部とが分離できる符号で、符号中に情報データの成分がそのままあらわれるため、データを扱いやすい。また、巡回符号は、符号に周期性があるため、復号時は一定の規則に従って順に検索できるため、誤り検索がシフトレジスタとバッファを用いて簡単に行える。

3

【0005】次に、組織符号の生成方法と復号方法について説明する。データ多項式 $D(x)$ 、冗長多項式 $R(x)$ 、符号生成多項式 $G(x)$ 、誤り多項式 $E(x)$ を以下のように定義する。

【0006】データ多項式 $D(x)$ は、数式(1)に示すよ

数1

$$D(x) = D_1 x^{k-1} + D_2 x^{k-2} + \dots + D_{k-1} x + D_k \quad \dots \text{数(1)}$$

【0008】冗長多項式 $R(x)$ は、数式(2)に示すよう ※とすると、最低 $2t$ バイトの冗長バイトが必要となる。に、 s バイト目の冗長バイト R_s を、 x の $(2t-s)$ 次項の係数とした x の多項式とする。 $R(x)$ は、訂正能力を t バイ※

数2

$$R(x) = R_1 x^{2t-1} + R_2 x^{2t-2} + \dots + R_{2t-1} x + R_{2t} \quad \dots \text{数(2)}$$

【0010】符号生成多項式 $G(x)$ は、数式(3)に示すように、 $D(x)$ を、符号語多項式 $W(x)$ に符号化するための x の多項式とする。訂正能力を t バイトとすると最低 $2t$ 次

数3

$$G(x) = (x - \alpha^{2^{t-1}})(x - \alpha^{2^{t-2}}) \dots (x - 1) \quad \dots \text{数(3)}$$

$$= \alpha^{n_1} x^{2^t} + \alpha^{n_2} x^{2^{t-1}} + \dots + \alpha^{n_{2t-1}} x + \alpha^{n_{2t}}$$

【0012】 $G(x)$ の根 α には数式(4)に示すような巡回性がある。

【0013】

【数4】

数4

$$\alpha^{N-1} = 1 \quad \dots \text{数(4)}$$

【0014】ここで、 N は、 $G(x)$ が含まれるガロア体の元の数で、 $G(x)$ によって作られる符号の最大シンボル(=1訂正単位)長は $(N-1)$ シンボルとなる。この巡

数5

$$W(x) = D(x)x^{2t} + R(x) \quad \dots \text{数(5)}$$

【0017】この時、 $R(x)$ は、数式(6)のようにな る。ここで、「 $A \bmod B$ 」は、 A を B で割った時の剰余多項式である。

数6

$$R(x) = D(x)x^{2t} \bmod G(x) \quad \dots \text{数(6)}$$

【0019】

【数7】

数7

$$W(x) \bmod G(x) = 0 \quad \dots \text{数(7)}$$

50 【0020】次に、RS符号の復号について説明する。

4

*うに、 s バイト目の情報バイト D_s を、 x の $(k-s)$ 次項の係数とした x の多項式とする。

【0007】

【数1】

【0009】

【数2】

★の多項式を用いる必要がある。

【0011】

【数3】

☆回性を利用して、 $N = (2 \text{ の } n \text{ 乗})$ とおくと、 $W(x)$ 、 $D(x)$ 、 $G(x)$ の係数は、 $G(x)$ の根 α のべき乗の形と、最大 $(n-1)$ 次の α 多項式、つまり n ビットのビット列の形の両方で記述することが出来る。

【0015】次に、RS符号の符号化について説明する。RS符号の符号化は、数式(5)に示すように、 $W(x)$ が $G(x)$ で割り切れ、かつ、 $R(x)$ と $D(x)$ の係数の次数が重ならないように、 $D(x)$ を x の $2t$ 乗して符号化する。

【0016】

【数5】

40 ◆ 【0018】

【数6】

5

6

記録媒体から読取った符号を $Y(x)$ 、誤り位置多項式を $E(x)$ とおく。 $Y(x)$ は、数式(8)に示すように、読み取った n バイトのデータの s バイト目の情報バイト Y_s を、

数8

$$Y(x) = Y_1x^{n-1} + Y_2x^{n-2} + \dots + Y_{n-1}x + Y_n \quad \dots \text{数}(8)$$

【0022】 $E(x)$ は、数式(9)に示すように、誤りバイトの位置とパターンを示す x の多項式とする。 i, j, \dots バイトにビットパターン E_i, E_j, \dots の誤りが発生したとする。 $Y(x), E(x)$ の係数も $W(x), D(x)$ と同じく、 n ビットのビット列で表現する事が出来る。

【0023】

【数9】

数9

$$E(x) = E_i x^i + E_j x^j + \dots \quad \dots \text{数}(9)$$

数10

$$Y(x) \bmod G(x) = E(x) \bmod G(x)$$

$$E(x) \bmod G(x) \begin{cases} = 0 & (\text{誤りが無い場合}) \\ = E(x) \bmod G(x) & (\text{誤りがある場合}) \end{cases}$$

…数(10)

【0026】誤りが1バイトの場合は、数式(10)より数式(11)を得て、 x に α のべき乗を順に代入することで簡単に E_i, i を求められる。

【0027】

【数11】

数11

$$E(x) \bmod G(x) = E_i x^i \quad \dots \text{数}(11)$$

数12

$$E(x) \bmod G(x) = E_i x^i + E_j x^j + \dots \quad \dots \text{数}(12)$$

【0030】

※ ※ 【数13】

数13

$$s_0 = E_i + E_j + \dots$$

$$s_1 = E_i \alpha^i + E_j \alpha^j + \dots$$

$$s_2 = E_i \alpha^{2i} + E_j \alpha^{2j} + \dots = (s_1)^2$$

$$s_3 = E_i \alpha^{3i} + E_j \alpha^{3j} + \dots$$

…数(13)

【0031】

50 【数14】

* x の $(n-s)$ 次の項の係数とした x の多項式とする。

【0021】

【数8】

【0024】この時、 $Y(x) \bmod G(x)$ を計算すると、誤り多項式 $E(x)$ の値によって数式(10)の関係が成り立つ。

【0025】

【数10】

30

【0028】しかし、誤りが2バイト以上の場合、 $E(x) \bmod G(x)$ は、数式(12)に示すようになる。このため、数式(12)に、 $G(x)$ の根(α の n 乗)($n=0, 1, \dots, 2t-1$)を代入して得られる $2t$ 本の方程式 $S(\alpha \text{ の } n \text{ 乗}) = S_n$ (数式(13))を用いて誤り位置と誤りパターンを検索する。この検索には数式(14)のように定義される誤り位置多項式 $\sigma(x)$ を用いる。

【0029】

【数12】

$$\sigma(x) = (1 - \alpha^i x)(1 - \alpha^j x) \dots \dots \text{数(14)}$$

【0032】数式(13)から数式(14)を求め、数式(14)に、 $x = \alpha$ の $-i$ 乗($i=0, 1, \dots, n-1$)を順に代入して i を求める。求めた i を数式(13)に代入して Ei を求める。

【0033】次に、積符号について説明する。積符号は、データを $n1$ 行 \times $n2$ 列のマトリクスと考え、各行、各列ごとにそれぞれ符号化を行い、復号時に各行の復号結果を利用して、各列の誤り訂正能力を向上させる方法である。ここで、最初に符号化を行う方を外符号C1、後に符号化を行う方を内符号C2とする。C1、C2の訂正能力をそれぞれ、 $t1$ バイト、 $t2$ バイトとすると、積符号の訂正能力は、最大で $\lfloor (d-1) / 2 \rfloor$ バイトまで*

数15

$$a_i = \frac{d_2 - 2\tau_i}{d_2} \quad \text{ただし}\tau\text{は訂正バイト数} \dots \text{数(15)}$$

【0036】次に、外符号C1について、誤り位置 i の集合 F を定義する。 F の初期値は、空集合 ϕ とする。また、 F に含まれる誤り位置 i の総数を h とする。

【0037】次に、外符号C1について、 h を消失誤りにおいて、 $\lfloor (d1-h-1) / 2 \rfloor$ バイトまでの誤り訂正処理を行う。ここで、 $d1=2 \times t1+1$ である。訂正能力を超

数16

$$\sum_{i=0}^{n-1} a_i f(y_i, w_i) > n - d1$$

ただし y_i は訂正処理前の符号、 w_i は訂正処理後の符号で

$f(y_i, w_i)$ は $y_i = w_i$ の時1、 $y_i \neq w_i$ の時-1になる関数。

【0040】数式(16)が成立した場合は、復号処理を終了する。成立しない場合は、集合 F への消失位置の追加処理に進む。

【0041】次に、集合 F への消失位置追加処理を行う。 F に含まれない位置 i で、 a_i が最小のものをすべて追加し、誤り訂正処理に戻る。ただし、 h が $d1$ 以上となった場合は、訂正能力を超える誤りが発生したと判定し、復号処理を終了する。

【0042】

【発明が解決しようとする課題】記憶装置の記憶媒体として利用されるフラッシュメモリは、図21に示すように、通常のMOSトランジスタのコントロールゲート直下に、電荷を担持及び放出することができる浮遊ゲート

*訂正することが可能である。ここで、 $d=(2 \times t1+1)(2 \times t2+1)$ であり、 $\lfloor A \rfloor$ は A を超えない最大の整数を表す。積符号の復号方法にはいろいろあるが、ここでは、例としてレディ・ロビンソン法について説明する。

【0034】最初に内符号C2について、 $t2$ バイトまでの誤りを訂正する。この時、行列の i 行目のデータ信頼性情報を数式(15)のように定義する。ここで、 $d2=2 \times t1+1$ である。なお、訂正能力を超える誤りが検出された場合は、 $a_i=0$ とする。

【0035】

【数15】

※える誤りが検出された場合は集合 F への消失位置の追加処理に進む。

【0038】次に、誤り訂正処理後と誤り訂正処理前の符号について、数式(16)が成立するか判定を行う。

【0039】

【数16】

 \dots 数(16)

を有する。そして、浮遊ゲートに担持する電荷量を制御することで、図22に示すように、MOSトランジスタのドレイン-ソース間の電流の流れを制御するコントロール電圧の閾値(スレッシュホールド電圧) V_{th} を変化させ、その変化を電流値として読み取っている。従来のフラッシュメモリでは、メモリ1素子に対し1ビットを対応させていたため、素子のランダム不良によるデータ誤りは、1ビット誤りであった。

【0043】しかし、近年、フラッシュメモリに対する大容量化、低コスト化の要求からフラッシュメモリにおいて、メモリ1素子に対し、2ビット以上を対応させる多値記録技術が必要となってきた。このような多値記録技術においては、一つの素子に対し、4つ以上の V

9

thを設定することが必要となり、各Vth間の間隔が狭くなるため、メモリ素子からのデータ読み出しエラーは必然的に増加する。また、素子不良によりVthが固定された場合は、1素子に2ビット以上の誤りが発生する可能性が生じる。

【0044】このため、特開平3-5995号公報のように、ビット単位で誤り訂正を行う場合、十分なデータ信頼率を得られなくなるおそれがある。

【0045】また、特開平2-301226号公報や特開平7-202717号公報のように、符号化率を固定してデータを二重に符号化すると、メモリチップの記憶素子のデータ信頼率の変化に対応出来ず、メモリチップの歩留まり率を上げるために、符号化効率を低めに設定する場合等、多値記録化による記憶容量の増加を有効に利用できない可能性がある。

【0046】本発明の目的は、十分なデータ信頼率が得られるとともに、メモリチップの記憶素子のデータ信頼率の変化に柔軟に対応することができる記憶装置を提供することにある。

【0047】

【課題を解決するための手段】本発明に係る記憶装置は、データを記憶する記憶手段と、当該記憶手段に対するデータの読み書きを制御する制御手段とを備えた記憶装置である。そして、前記記憶手段は、データを格納するメモリ手段と、当該メモリ手段に書き込むデータの誤り訂正符号化を行い、また、前記メモリ手段から読み出したデータの誤り検出及び誤り訂正を行う内部ECC手段とを備え、当該内部ECC手段によって行った誤り訂正に関する信頼性情報を出力し、前記制御手段は、前記記憶手段に書き込むデータの誤り訂正符号化を行い、また、前記記憶手段から読み出したデータの誤り検出及び誤り訂正を行う外部ECC手段を備え、前記外部ECC手段は、前記記憶手段から出力された信頼性情報を利用して誤り訂正を行うことを特徴とする。

【0048】この場合、前記制御手段は、ホストから書き込まれたデータを、外部ECC手段によって、外符号化し、前記記憶手段は、前記制御手段によって書き込まれた外符号を、内部ECC手段によって、内符号化し、前記メモリ手段に書き込むようにしてもよい。

【0049】また、複数の動作モードを持ち、例えば、第1の動作モードでは、前記外部ECC手段のみによる符号化を行い、第2の動作モードでは、前記内部ECC手段のみによる符号化を行い、第3の動作モードでは、前記外部ECC手段及び内部ECC手段による積符号化を行うようにしてもよい。

【0050】本発明に係るコントローラは、データの記憶を行う記憶手段に対するデータの読み書きを制御するコントローラである。そして、前記記憶手段に書き込むデータを誤り訂正符号化し、また、前記記憶手段から読み出したデータの誤りの有無を検出し、誤りが検出され

10

た場合に当該データの誤りを訂正する外部ECC手段を有し、前記外部ECC手段は、前記記憶手段から出力された、誤り訂正に関する信頼性情報を利用して、読み出しデータの誤り訂正を行うことを特徴とする。

【0051】本発明に係るメモリチップは、データを格納するメモリ手段と、当該メモリ手段に書き込むデータの誤り訂正符号化を行い、また、前記メモリ手段から読み出したデータの誤り検出及び誤り訂正処理を行う内部ECC手段とを備え、当該内部ECC手段で行われた誤り訂正処理に関する信頼性情報を出力することを特徴とする。

【0052】この場合において、前記内部ECC回路は、その動作モードに応じて、誤り訂正能力を変更できるようにしてもよい。

【0053】また、以上の場合において、前記内部ECC手段及び外部ECC手段は、誤り訂正符号としてシンボル単位訂正符号で、組織符号となる巡回符号（例えば、リードソロモン符号）を用いるようにしてもよい。

【0054】また、前記内部ECC手段及び外部ECC手段は、それぞれ、データの誤り訂正符号化を行う誤り訂正符号生成手段と、データの誤りの有無を検出する誤り検出手段と、データに誤りが検出された場合に、当該データの誤りを訂正する誤り訂正手段とを備えるようにしてもよい。

【0055】また、前記信頼性情報は、ある特定の規則によって重み付けがなされるようにしてもよい。

【0056】

【発明の実施の形態】以下、図面を参照しつつ、本発明の実施の形態について詳細に説明する。

【0057】図1は、本発明によるデジタル情報記憶装置の概要を示す図である。同図に示すように、本発明によるデジタル情報記憶装置101は、メモリチップ102とインターフェースLSI106とを備え、システムバス109に接続される。デジタル情報記憶装置101は、システムバス109を介して、ホスト等とデジタル情報の受け渡しを行い、ホストから書き込まれたデジタル情報を記憶する。

【0058】メモリチップ102は、メモリ103と内部ECC回路104とを備える。メモリ103は、半導体を利用した記憶媒体で、データの記憶を行う。内部ECC回路104は、メモリチップ102に書き込まれたデータのECC符号化や、メモリ103から読み出されたデータの誤り検出・訂正を行う。また、メモリチップ102は、内部ECC回路104で行った誤り訂正に関する信頼性情報をチップ外部に出力する。

【0059】インターフェースLSI106は、システムバス109とのインターフェース制御を行う。また、インターフェースLSI106は、外部ECC回路1061を備える。外部ECC回路1061は、システムバス109を介して、ホスト等から渡されたデータのECC

11

C符号化及びメモリチップ102から読み出されたデータの誤り検出・訂正を行う。外部ECC回路1061は、その動作モードに応じて、内部ECC回路104から出力される信頼性情報を利用して、誤り訂正を行う。

【0060】図2は、本発明によるデジタル情報記憶装置のより具体的な構成を示す図である。本記憶装置は、記憶媒体として、フラッシュメモリを用いたものであり、フラッシュメモリカード（PC-ATAカード）として実現したものである。

【0061】同図に示すように、フラッシュメモリカード501は、メモリチップ102と、インターフェースLSI106と、マイコン504と、DRAM505とを備える。各構成要素は、制御信号線、ローカルアドレスバス、ローカルデータバスによって接続されている。

【0062】メモリチップ102は、メモリ103と内部ECC回路104を含むICチップである。インターフェースLSI106は、システムバス109とのインターフェース制御を行うLSIであり、外部ECC回路1061を備える。

【0063】マイコン504は、フラッシュメモリカード501のコントローラの役割を担っており、中央処理装置（CPU）5041と、ROMと、RAMとを備える。CPU5041は、システムバス109を介して、ホストから送られてきた命令を解釈し、解釈結果に応じて、メモリチップ102に対するデータの読み書きや、DRAM505に対するデータの読み書きを制御する。

【0064】DRAM505は、ホストとメモリチップ102との間でデータのやり取りを行う際、データのバッファの役目を担う補助メモリである。

【0065】図3は、図2に示したフラッシュメモリカード501の具体的な実装イメージを示す図である。フラッシュメモリカード501は、I/Fコネクタ201を介して、例えば、ノート型PCのPCカード・スロットに装着されて、記憶装置として利用される。

【0066】次に、メモリチップ102の構成について説明する。図4は、メモリチップ102の構成を示す図である。同図に示すように、メモリチップ102は、メモリ103と、デコーダ602と、内部コントローラ604と、スイッチ／セクタ603と、センスアンプ・ラッチ605と、内部ECC回路104と、I/Oバッファ601とを備える。図4は、メモリバンクが2つある場合を示しており、各バンク毎に、デコーダ602とセンスアンプ・ラッチ605を備えている。

【0067】メモリ103は、データの記憶を行う記憶媒体である。I/Oバッファ601は、メモリチップ102の外部からローカルデータバスを介して送られてきたデータや、内部ECC回路104から送られてきたデータのバッファの役目を担う補助メモリである。デコーダ602は、データの読み書きを行う際、ローカルアドレスバスを介して渡されるアドレスをデコードし、メモ

12

リ103上でのデータアクセス位置を制御する。スイッチ／セクタ603は、メモリ103へのデータ書き込み時あるいはメモリ103からのデータ読み出し時に、内部コントローラ604からの制御信号に応じて、メモリ103の各バンクと内部ECC回路104との間のデータ入出力の切り替えを行う回路である。

【0068】内部コントローラ604は、メモリチップ102内部の各構成要素の制御を行う回路である。センスアンプ・ラッチ605は、メモリ103に対するデータの読み書き時に、読み出しデータの検出及び増幅やデータの保持を行う。

【0069】内部ECC回路104は、メモリ103に書き込むデータの符号化を行い、また、メモリ103から読み出したデータの誤り検出及び誤り訂正を行う。

【0070】次に、内部ECC回路104及び外部ECC回路1061について説明する。

【0071】本実施形態では、内部ECC回路104と外部ECC回路1061で共有して用いるECCにRS符号を用いる。なお、データの誤り訂正、検出に用いるECCは、これに限られず、シンボル単位訂正符号で、組織符号となる巡回符号であれば何を用いても構わない。

【0072】また、内部ECC回路104の訂正能力は、2バイト、外部ECC回路1061の訂正能力は、5バイトとする。なお、内部ECC回路104の訂正能力をaバイト、外部ECC回路1061の訂正能力をbバイトとすると、a、bは、 $a \leq b$ なる整数であれば、どのような値でもよい。

【0073】また、本実施形態では、システムバスを介したホスト等からの情報データの読み込み／書き込み単位は、512バイトとし、用いるガロア体は、512バイトの情報データを一括して変換でき、1シンボルを1バイトに対応させることができる最小のガロア体という条件から、2の10乗とする。すなわち、1シンボルのビット数は、10ビットである。

【0074】この場合、情報データ部については、各ECC回路内部では、8ビットのデータに、例えば、2ビットの0をつけて、10ビットのシンボルとして扱い、実際にメモリ103に格納するときは、8ビットのまま記録する。一方、冗長データ部を格納する際には、バイト単位で情報を記録するとしての最小バイト数で書き込むものとする。例えば、4シンボル（＝40ビット）の冗長データ部は、5バイトの領域を使って格納する。なお、冗長データ部の格納方法は、これに限られず、例えば、各シンボルを2バイトの領域に格納するようにしてもよい。以下では、シンボルという言葉を用いる。

【0075】本実施形態は、動作モード制御信号sig1で指定される動作モードによって、外部ECC回路10

13

61及び内部ECC回路104の符号化時及び復号時の動作が異なる。なお、動作モード制御信号sig₁の値は、例えば、特定のレジスタの設定や、ジャンパーピン¹⁰の設定で指定される。

【0076】図5は、動作モード制御信号sig₁と各ECC回路104、1061の動作との対応関係を示す表である。この場合、動作モード制御信号sig₁は、4ビットの信号で、図5の表の示した値以外の値は用いないこととする。

【0077】図5のECC回路の動作の欄で、1行で表¹⁰記されている場合は、符号化時と復号時とで、訂正能力（符号化の単位となるバイト長および付加する冗長シンボル数）及び訂正処理が一致していることを示す。一方、上下二段に表記されている箇所は、符号化時と復号時とで各ECC回路における動作が異なることを示す。この場合、上段が符号化時の動作、下段が復号時の動作を示す。

【0078】同図に示すように、動作モード制御信号sig₁=「0010」の場合、内部ECC回路104は、動作せず、外部ECC回路1061は、512バイトの情報²⁰データに対して、5バイトまでの誤り訂正可能な誤り訂正符号化及び誤り訂正処理を行う。

【0079】動作モード制御信号sig₁=「1111」の場合は、積符号による誤り訂正符号化及び復号を行う。積符号では、情報データが正方マトリクスデータに近いほど符号化効率が良いため、本実施形態では、512バイトの情報データを、16バイト×32バイトのマトリクスデータとして扱う。すなわち、内部ECC回路104は、16バイトの情報データに対して、2バイトまでの誤り訂正可能な誤り訂正符号化及び誤り訂正処理を行³⁰う。また、外部ECC回路1061は、32バイトの情報データに対して、5バイトまでの誤り訂正可能な誤り訂正符号化及び誤り訂正処理を行う。

【0080】動作モード制御信号sig₁=「1000」の場合は、内部ECC回路104は、512バイトの情報データに対して、2バイトまでの誤り訂正可能な誤り訂正符号化及び誤り訂正処理を行う。この場合、外部ECC回路1061は、動作を行わない。

【0081】動作モード制御信号sig₁=「0100」の場合は、内部ECC回路104は、符号化時、512バ⁴⁰イトの情報データに対して、5バイトまでの誤り訂正可能な誤り訂正符号化を行う。一方、復号時には、512バイトの情報データ及び10シンボルの冗長データに対して、内部ECC回路104で可能な2バイトまでの誤り訂正処理を行う。なお、内部ECC回路104では、5バイトまでの誤り検出は可能である。また、この場合も、外部ECC回路1061は、動作を行わない。

【0082】動作モード制御信号sig₁=「0110」の場合は、内部ECC回路104は、符号化時、512バ⁵⁰イトの情報データに対して、5バイトまでの誤り訂正可能

14

な誤り訂正符号化を行う。一方、復号時には、動作を行¹⁰わない。また、外部ECC回路1061は、符号化時、動作を行わず、復号時に、512バイトの情報データ及び10シンボルの冗長データに対して、5バイトまでの誤り訂正可能な誤り訂正処理を行う。すなわち、動作モード制御信号sig₁=「0110」の場合、誤り訂正符号化は、内部ECC回路104で行い、誤り訂正処理は、外部ECC回路1061で行う。

【0083】次に、外部ECC回路1061の構成について説明する。図6は、外部ECC回路1061の構成を示す図である。

【0084】同図に示すように、外部ECC回路1061は、8-10変換器1701と、シンドローム/符号生成手段1702と、誤り訂正処理開始判定手段1703と、誤り評価式生成手段1704と、誤り位置・誤り値検索手段1705と、消失位置計算手段1706と、訂正不能誤り検出手段1707と、誤り訂正手段1708と、10-8変換器1709とを備える。

【0085】8-10変換器1701は、外部ECC回路1061に²⁰入力される8ビットのデータを10ビットのシンボルに変換する。例えば、情報データ部²⁰の場合は、2ビットの0を追加する。また、冗長データの場合は、例えば、2つのバイトから、10ビットのシンボルを抽出する。

【0086】シンドローム/符号生成手段1702は、ホストから渡されたデータを符号化し、また、メモリチップ102から読み出したデータからシンドロームを生成する。シンドローム/符号生成手段1702の詳細については、後述する。

【0087】誤り訂正処理開始判定手段1703は、符号化時には、シンドローム/符号生成手段1702によって生成された符号を10-8変換器1709に出力し、復号時には、誤り訂正処理の開始を判定する。

【0088】誤り評価式生成手段1704は、シンドロームまたは行データ信頼性情報からデータ誤り個数を判定する。誤り位置・誤り値検索手段1705は、誤り評価式生成手段1704によって生成された情報から、データ誤りの位置および誤り値の検索を行う。

【0089】消失位置計算手段1706は、内部ECC回路104から渡される行データ信頼性情報から消失位置を計算する。訂正不能誤り検出手段1707は、データ誤りの位置および誤り値を用いて、訂正能力を超えた誤りを検出する。

【0090】誤り訂正手段1708は、誤り位置・誤り値検索手段1705から得られた誤り位置、誤り値を用いて、メモリチップ102から読み出したデータの誤りを訂正する。例えば、誤り位置のシンボルに対して、誤り値をXORすることで訂正を行う。

【0091】10-8変換器1709は、10ビットのシンボルを8ビットデータに変換する。例えば、情報デ

15

ータ部の場合は、8-10変換器1701で追加した2ビットを削除する。また、冗長データ部の場合は、10ビットのシンボルを、2つのバイトに分けて出力する。

【0092】次に、前述したシンドローム／符号生成手段1702の詳細について説明する。図7は、シンドローム／符号生成手段1702の構成を示す図である。

【0093】同図に示すように、シンドローム／符号生成手段1702は、シフトレジスタ部2101と、ビット乗算部2102と、スイッチSW11、SW12、SW13とを備える。

【0094】シフトレジスタ部2101は、10ビットのフリップフロップDが10段構成になったものである。ビット乗算部2102は、入力されたデータに次数に応じた定数を乗じるビット乗算器gが10段構成になったものである。スイッチSW11とSW12は、オンオフスイッチで、スイッチSW13は、a、bの2接点を持つ3点スイッチである。

【0095】次に、内部ECC回路104の構成について説明する。図8は、内部ECC回路104の構成を示す図である。

【0096】同図に示すように、内部ECC回路104は、8-10変換器1801と、シンドローム／符号生成手段1802と、誤り訂正処理開始判定手段1803と、誤り評価式生成手段1804と、誤り位置・誤り値検索手段1805と、訂正不能誤り検出手段1806と、行データ信頼性情報生成手段1807と、誤り訂正手段1808と、10-8変換器1809とを備える。

【0097】8-10変換器1801は、前述した8-10変換器1701と同様にして、入力された8ビットデータを10ビットのシンボルに変換する。

【0098】シンドローム／符号生成手段1802は、メモリ103に書き込むデータを符号化し、また、メモリ103から読み出したデータからシンドロームを生成する。シンドローム／符号生成手段1802の詳細については、後述する。

【0099】誤り訂正処理開始判定手段1803は、符号化時には、符号化されたデータを、10-8変換器1809に出力し、復号時には、誤り訂正処理の開始の判定をする。

【0100】誤り評価式生成手段1804は、シンドロームからデータ誤り個数を判定する。誤り位置・誤り値検索手段1805は、誤り評価式生成手段1804によって生成された情報から、データ誤りの位置および誤り値の検索を行う。

【0101】訂正不能誤り検出手段1806は、データ誤りの位置および誤り値を用いて、訂正能力を超えた誤りを検出する。行データ信頼性情報生成手段1807は、誤り位置・誤り値検索手段1805で得られた誤り位置情報から行データ信頼性情報を生成する。

【0102】誤り訂正手段1808は、誤り位置・誤り

16

値検索手段1805が出力する誤り位置、誤り値を用いて、メモリ103から読み出したデータの誤りを訂正する。10-8変換器1809は、前述した10-8変換器1709と同様にして、10ビットのシンボルを8ビットデータに変換する。

【0103】次に、前述したシンドローム／符号生成手段1802の構成について説明する。

【0104】図9は、シンドローム／符号生成手段1802の構成を示す図である。同図に示すように、シンドローム／符号生成手段1802は、低次側シンドローム生成器16Aと、高次側シンドローム生成器16Bと、データ出力部16Cと、セクタ部16Dとを備える。各構成要素の接続形態は、動作モード制御信号sig_1の値によって変化する。

【0105】低次側シンドローム生成器16Aは、シフトレジスタ部1601と、ビット乗算部1602と、スイッチSW1～SW3とを備える。シフトレジスタ部1601は、10ビットのフリップフロップDが4段構成になったものである。ビット乗算部1602は、入力されたデータに次数に応じた定数を乗じるビット乗算器gが4段構成になったものである。スイッチSW1～SW3は、オンオフスイッチである。

【0106】高次側シンドローム生成器16Bは、シフトレジスタ部1604と、ビット乗算部1605と、スイッチSW5、SW7とを備える。シフトレジスタ部1604は、10ビットのフリップフロップDが6段構成になったものである。ビット乗算部1605は、入力されたデータに、次数に応じた定数を乗じるビット乗算器gが6段構成になったものである。スイッチSW5、SW7は、オンオフスイッチである。

【0107】データ出力部16Cは、a、bの2接点を持つ3点スイッチSW6を備える。

【0108】セクタ部16Dは、動作モード制御信号sig_1の値によって、低次側シンドローム生成器16Aと、高次側シンドローム生成器16Bと、データ出力部16Cとの間の接続を変更する。セクタ部16Dは、セクタ1～5と、インバータINVとを備える。

【0109】セクタ1及びセクタ2は、2入力1出力セクタで、動作モード制御信号sig_1の最上位ビットの値が「1」の時、黒丸側を選択し、「0」の時、白丸側を選択するセクタである。

【0110】セクタ3、セクタ4及びセクタ5は、1入力1出力ゲートで、セクタ信号線2の値が「1」の時、すなわち、動作モード制御信号sig_1の最上位ビットが「0」の時、入力されたデータをそのまま出力し、セクタ信号線2の値が「0」の時、すなわち、動作モード制御信号sig_1の最上位ビットが「1」のとき、入力されたデータを出力しない。

【0111】低次側シンドローム生成器16A及び高次側シンドローム生成器16Bは、外部ECC回路106

17

1のシンドローム／符号生成手段1702と、符号生成時のガロア体の元を同一にするためフリップフロップDのビット数を同一にし、符号生成多項式が同一になるように、ビット乗算部の各定数gを定めている。

【0112】次に、以上説明したような構成を有するフラッシュメモリカード501におけるデータ書き込み時の処理について説明する。

【0113】まず、データ書き込み時のデータパスについて説明する。

【0114】図10は、動作モード制御信号sig_1の値10に応じたデータパスの変化の様子を概念的に示す図である。

【0115】前述したように、動作モード制御信号sig_1の値が「0010」の場合、内部ECC回路104は使用されない。従って、データパスは、1、2、4、6、8の順になる。すなわち、ホスト等からフラッシュメモリカード501に書き込まれたデータは、まず、外部ECC回路1061に渡され、外部ECC回路1061によって符号化され、その後、内部ECC回路104によって符号化されることなく、メモリ103に書き込まれる。20

【0116】また、動作モード制御信号sig_1の値が「1111」の場合、データパスは1、2、4、5、7、8となる。すなわち、ホスト等からフラッシュメモリカード501に書き込まれたデータは、まず、外部ECC回路1061に渡され、外部ECC回路1061によって、外符号に符号化される。その後、内部ECC回路104に渡され、内部ECC回路104によって内符号に符号化され、メモリ103に書き込まれる。

【0117】動作モード制御信号sig_1の値が「1000」、「0100」、「0110」の場合、データパスは1、3、5、7、8となる。すなわち、ホスト等からフラッシュメモリカード501に書き込まれたデータは、外部ECC回路1061によって符号化されることなく、メモリチップ102に書き込まれ、内部ECC回路104によって符号化された後、メモリ103に書き込まれる。

【0118】次に、データ書き込み時の符号化処理の詳細について説明する。

【0119】図11は、データ書き込み時の符号化処理40の流れを示す図である。

【0120】フラッシュメモリカード501では、前述したように、その動作モードによって行う符号化処理が異なるので、まず、動作モード制御信号sig_1の値が「1111」であるか否かを判別する(S1101)。

【0121】その結果、動作モード制御信号sig_1の値が「1111」の場合(S1101: YES)、512バイトのデータを積符号化(外符号化)するため、外部ECC回路1061に入力する前に、情報データの並べ替えを行う(S1102)。前述したように、本実施形態では、512 50

18

バイトの情報データを、16バイト×32バイトのマトリクスデータとして扱う。外部ECC回路1061は、適宜並べ替えて入力される512バイトのデータを、32バイトごとに符号化する(S1104)。情報データの並べ替え及び符号化処理の詳細については、後述する。

【0122】一方、動作モード制御信号sig_1の値が「1111」でない場合は(S1101: NO)、続けて、動作モード制御信号sig_1の値が「0010」であるか否かを判定する(S1103)。その結果、動作モード制御信号sig_1の値が「0010」の場合(S1103: YES)、外部ECC回路1061は、入力される512バイトの情報データを、5バイトまでの誤り訂正が可能な(512+10)シンボルの符号に符号化する(S1104)。ここでの外部ECC回路1061による符号化処理の詳細については、後述する。

【0123】一方、動作モード制御信号sig_1の値が「0010」でない場合は(S1103: NO)、すなわち、「1000」、「0100」又は「0110」の場合、512バイトのデータは、外部ECC回路1061による符号化を行うことなしに、そのまま内部ECC回路104に渡される。

【0124】以上のようにして、メモリチップ102外部での処理が終了すると、続いて、メモリチップ102内部での符号化処理に移る。メモリチップ102内部の符号化処理もその動作モードによって異なるので、まず、動作モード制御信号sig_1の値が「0010」であるか否かの判定を行う(S1105)。

【0125】その結果、動作モード制御信号sig_1の値が「0010」の場合(S1105: YES)、内部ECC回路104では符号化を行わず、例えば、メモリチップ102に、バイト単位でシーケンシャルに書き込まれる512バイトの情報データ+10シンボルの冗長データを、メモリ103にそのまま書き込み(S1107)、データ書き込み処理が終了する。

【0126】一方、動作モード制御信号sig_1の値が「0010」でない場合は(S1105: NO)、内部ECC回路104は、その動作モードに応じた内部符号化を行う(S1106)。ここでの内部ECC回路104による符号化処理の詳細については、後述する。内部ECC回路104によって符号化された情報データは、メモリ103に書き込まれ(S1107)、データ書き込み処理が終了する。

【0127】次に、外部ECC回路1061及び内部ECC回路104による符号化処理の詳細について説明する。

【0128】まず、外部ECC回路1061及び内部ECC回路104によって、512バイトの情報データを積符号化する場合(動作モード制御信号sig_1=「1111」の場合)について説明する。

【0129】図12は、本実施形態で採用する積符号の構成の概念を示す図である。同図に示すように、512バイトの情報データを、16バイト×32バイトのマト

19

リクスデータとして考え、外符号C1は、32バイトの列データを符号化し、内符号C2は16バイトの行データを符号化する。符号化する情報量は、1行あたり $16 \times 8 = 128$ ビット、1列あたり $32 \times 8 = 256$ ビットになる。

【0130】また、外部ECC回路1061の訂正能力は5シンボルだから、1つの列データの符号化に必要な冗長シンボル長は、10シンボル($= 10 \times 10 \div 8 = 13$ バイト)になる。また、内部ECC回路104の訂正能力は、2シンボルだから、1つの行データの符号化に必要な冗長シンボル長は4シンボル($4 \times 10 \div 8 = 5$ バイト)になる。

【0131】前述したように、本実施形態では、冗長シンボルは、バイト単位で情報を記録するとしての最小バイト数で書き込むものとするので、外符号C1の冗長データ部R1の格納には、 $13 \times 16 = 208$ バイトの領域が必要となる。また、内符号C2の冗長データ部R2の格納には、 $5 \times 42 = 210$ バイトの領域が必要となる。

【0132】実際にメモリ103に格納されるときは、²⁰ 情報データは、512バイトの1次元配列データとして格納されるため、マトリクスデータへの変換は必ずしも必要ではなく、1次元配列データのままで扱ってもよい。以下の説明では情報データを1次元配列データとして扱う。

【0133】図13は、512バイトの1次元配列データから外符号C1の情報データ部を抽出し、冗長データ部を生成する方法を示す図である。ホストからフラッシュメモリカード501に書き込まれた512バイトのデータは、一旦、DRAM505に格納される。そして、³⁰ マイコン504は、まず、先頭の1バイトのデータを読み出し、外部ECC回路1061に入力する。次に、データアドレスを15バイトスキップして、1バイトのデータを読み出し、外部ECC回路1061に入力する。このように、16バイト毎のデータを順次読み出して、読み出したデータを順次、外部ECC回路1061に入力する。これを32回繰り返し、32バイトの列データを外部ECC回路1061に入力する。32バイトの列データの inputs が終了すると、今度は、この32バイトの列データに対して生成された10シンボルの冗長データ⁴⁰を外部ECC回路1061から読み出して、これをDRAM505の所定の領域、例えば、情報データ部の後ろに格納する。

【0134】以上の操作が終了すると、今度は、前から2バイト目から、16バイト毎にデータを読み出し、以上と同様の操作を行う。このような処理を16回、すなわち、図12に示したマトリクスの第1列～第16列について行い、10シンボルの冗長データを16個分生成する。

【0135】以上のようにして、外部ECC回路106⁵⁰

20

1は、512バイトのデータを32バイトごとに符号化する。

【0136】次に、この時の外部ECC回路1061内の動作について説明する。

【0137】外部ECC回路1061に入力された8ビットのデータは、まず、8-10変換器1701により、10ビットに変換されたあと、シンドローム/符号生成手段1702に入力される。

【0138】このとき、図7に示したシンドローム/符号生成手段1702では、スイッチSW11、SW12がオンにされ、SW13のb側が選択されている。このような状態のシンドローム/符号生成手段1702において、情報データは、1シンボルずつ、入力端子から、スイッチSW11、SW12を介して、ビット乗算部2102及びシフトレジスタ部2101に順次入力される。また、情報データは、それと同時に、スイッチSW13を通して、そのまま出力端子に出力される。そして、32バイトすべての情報データの inputs が終了すると、シフトレジスタ部2101に、10段 \times 10ビットの冗長データが保持されていることになる。以上のようにして、32バイトの情報データに対して、10シンボルの冗長データが生成される。

【0139】次に、以上のようにして生成された10シンボルの冗長データを出力端子から出力する。そのため、スイッチSW13のa側を選択し、スイッチSW11、SW12を、オフとする。このとき、ビット乗算部2102には、0が入力されるため、シフトレジスタ部2101の各フリップフロップDの値、すなわち、冗長データは、そのまま高次の項からSW13を通して出力される。

【0140】シンドローム/符号生成手段1702から出力された10ビットのデータは、10-8変換器1709で、8ビットのデータに変換された後、外部ECC回路1061の外部に出力される。

【0141】以上の操作が16回繰り返されることで、512バイトのデータから 42×16 シンボルの外符号が生成される。

【0142】以上のようにして外符号化が終了すると、次に、生成された外符号のメモリチップ102に対する書き込みが行われる。この外符号のメモリチップ102への書き込みでは、冗長データ部は、10シンボル \times 16個からなるR1を16シンボル \times 10個のデータへ変換して書き込まれる。

【0143】図13の場合、冗長データ部R1において、1シンボル読み取るごとにデータアドレスを9シンボルずつスキップする。これを16回繰り返し16シンボルの列データを生成する。この操作を10回繰り返し、10個の列データを生成する。

【0144】以上のようにしてメモリチップ102に書き込まれる 16×42 シンボルのデータは、8ビット毎

21

に、内部ECC回路104に順次入力され、16シンボル単位で、内符号C2に符号化される。

【0145】次に、この時の内部ECC回路104内の動作について説明する。

【0146】内部ECC回路104に入力された8ビットのデータは、まず、8-10変換器1801で10ビットのシンボルに変換された後、シンドローム/符号生成手段1802に入力される。

【0147】このとき、図9に示したシンドローム/符号生成手段1802では、セクタ部16Dによって、¹⁰端子1と端子9、端子3と端子10が接続され、低次側シンドローム生成器16Aとデータ出力部16Cが接続される。また、スイッチSW1、SW2、SW3がオンとされ、スイッチSW6は、a側が選択される。

【0148】このような状態において、情報データは、1シンボルずつ、スイッチSW1、SW2を介して、ビット乗算部1602及びシフトレジスタ部1601に順次入力されると同時に、スイッチSW6を通して、出力端子に出力される。16バイトの情報データの入力が終了すると、シフトレジスタ部1601に、4段×10ビ²⁰ットの冗長データが保持されていることになる。

【0149】次に、この4シンボルの冗長データをシンドローム/符号生成手段1802から出力する。そのため、スイッチSW6のb側を選択し、スイッチSW1、SW2、SW3は、オフとする。このとき、ビット乗算部1602には、0が入力されるため、シフトレジスタ部1601の各フリップフロップDの値、すなわち、冗長データは、そのまま高次の項からスイッチSW6を通して出力される。

【0150】シンドローム/符号生成手段1802から³⁰出力された10ビットのデータは、10-8変換器1809で、8ビットのデータに変換された後、内部ECC回路104の外部に出力され、メモリ103に書き込まれる。

【0151】以上のような操作を(32+10)回繰り返すことで、内符号C2の生成が終了する。

【0152】次に、積符号化以外の符号化処理の詳細について説明する。

【0153】まず、外部ECC回路1061によって、512バイトの情報データを、5バイトまでの誤り訂正⁴⁰可能な(512+10)シンボルの符号に符号化する場合(動作モード制御信号sig_1=「0010」の場合)について説明する。

【0154】この時の外部ECC回路1061のシンドローム/符号生成手段1702における動作は、前述した積符号化時の場合(動作モード制御信号sig_1の値=「1111」の場合)とほぼ同様である。

【0155】すなわち、スイッチSW11、SW12がオンにされ、SW13のb側が選択されたシンドローム/符号生成手段1702において、情報データは、1⁵⁰

22

シンボルずつ、入力端子から、スイッチSW11、SW12を介して、ビット乗算部2102及びシフトレジスタ部2101に順次入力され、それと同時に、スイッチSW13を通して、そのまま出力端子に出力される。そして、512バイトの情報データの入力が終了すると、シフトレジスタ部2101に、10段×10ビットの冗長データが保持されていることになる。以上のようにして、512バイトの情報データに対して、10シンボルの冗長データが生成される。

【0156】次に、このようにして生成された冗長データを、前述した積符号化時の場合と同様にして、シンドローム/符号生成手段1702から出力する。

【0157】以上のようにして、512バイトの情報データから、(512+10)シンボルの符号が生成される。

【0158】次に、内部ECC回路104によって、512バイトのデータを2バイトまでの誤り訂正可能な(512+4)シンボルの符号に符号化する場合(動作モード制御信号sig_1=「1000」の場合)について説明する。

【0159】この場合の内部ECC回路104のシンドローム/符号生成手段1802における動作も、前述した積符号化時の場合とほぼ同様である。

【0160】すなわち、セクタ部16Dによって、低次側シンドローム生成器16Aとデータ出力部16Cが接続され、スイッチSW1、SW2、SW3がオンとされ、スイッチSW6のa側が選択されたシンドローム/符号生成手段1802において、情報データは、1シンボルずつ、スイッチSW1、SW2を介して、ビット乗算部1602及びシフトレジスタ部1601に順次入力されると同時に、スイッチSW6を通して、出力端子に出力される。512バイトすべての情報データの入力が終了すると、シフトレジスタ部1601に、4段×10ビットの冗長データが保持されていることになる。

【0161】次に、この4シンボルの冗長データを、前述したのと同様にして、シンドローム/符号生成手段1802から出力する。

【0162】以上のようにして、512バイトの情報データから、(512+4)シンボルの符号が生成される。

【0163】次に、内部ECC回路104によって、512バイトのデータを5バイトまでの誤り訂正可能な(512+10)シンボルの符号に符号化する場合(動作モード制御信号sig_1=「0100」または「0110」の場合)について説明する。

【0164】この場合、内部ECC回路104のシンドローム/符号生成回路1802は、右から低次側シンドローム生成器16A、高次側シンドローム生成器16B、データ出力部16Cの順に接続され、外部ECC回路1061のシンドローム/符号生成回路1702と同

23

様の構成になる。そして、動作モード制御信号sig₁の値が「0010」の場合の外部ECC回路1061と同様に、10シンボルの冗長データを生成し、512バイトの情報データを5バイト訂正可能な符号へ符号化する。

【0165】次に、データ読み出し時の処理について説明する。

【0166】まず、データ読み出し時のデータパスの変化について説明する。図14は、動作モード制御信号sig₁の値に応じたデータパスの変化の様子を概念的に示す図である。同図において、例えば、「**10」とある場合、「**」の部分は、任意の値を意味する。

【0167】まず、前述したように、動作モード制御信号sig₁の値が「0010」又は「0110」（すなわち、「**10」）の場合、内部ECC回路104による誤り検出・訂正処理は行わない。従って、データパスは、図14に示した8、6、4、2、1の順になる。すなわち、メモリ103から読み出されたデータは、内部ECC回路104で誤り検出・訂正処理が行われることなく、そのまま、メモリチップ102の外部に出力される。メモリチップ102の外部に出力されたデータは、外部ECC回路1061に入力され、誤り検出・訂正処理が行われる。外部ECC回路1061により誤り検出・訂正処理が行われたデータは、フラッシュメモリカード501の外部に渡される。

【0168】また、動作モード制御信号sig₁の値が「1000」又は「0100」（すなわち、「**00」）の場合、データパスは、8、7、5、3、1となる。すなわち、メモリ103から読み出されたデータは、内部ECC回路104に入力され、誤り検出・訂正処理が行われる。内部ECC回路104によって誤り検出・訂正が行われたデータは、メモリチップ102の外部に出力される。メモリチップ102から出力されたデータは、外部ECC回路1061で誤り検出・訂正処理が行われることなく、フラッシュメモリカード501の外部に出力される。

【0169】また、動作モード制御信号sig₁の値が「1111」の場合、データパスは、8、7、5、4、2、1となる。すなわち、メモリ103から読み出されたデータは、内部ECC回路104及び外部ECC回路1061の両方で、積符号による誤り検出・訂正処理が行われる。

【0170】次に、内部ECC回路104による復号処理について説明する。図15は、内部ECC回路104による復号処理の流れを示す図である。

【0171】メモリ103から読み出されたデータを内部ECC回路104で復号するか否かは、前述したように、その動作モードによるので、まず、動作モード制御信号sig₁が「**10」であるか否かを判別する（S1201）。その結果、動作モード制御信号sig₁の値が「**1

24

0」（具体的には、「0010」または「0110」）の場合（S1201: YES）、内部ECC回路104では復号を行わず、メモリ103から読み出されたデータ（情報データ+冗長データ）をそのまま外部ECC回路1061に出力する。

【0172】一方、動作モード制御信号sig₁の値が「**10」でない場合は（S1201: NO）、内部ECC回路104による復号処理を行う。符号の復号単位は、動作モード制御信号sig₁の値が「1000」の場合、（512+4）シンボル、「1111」の場合は、（16+4）シンボル、「0100」の場合、（512+10）シンボルである。

【0173】内部ECC回路104による復号を行う場合は、最初に、前述した復号単位について、S_nの値を計算する（S1203）。この計算は、シンドローム/符号生成手段1802で行われる。

【0174】このときのシンドローム/符号生成手段1802の回路構成は、動作モード制御信号sig₁の値が「1111」又は「1000」の場合は、右から低次側シンドローム生成器16A、データ出力部16Cの順に接続され、動作モード制御信号sig₁の値が「0100」の場合は、低次側シンドローム生成器16A、高次側シンドローム生成器16B、データ出力部16Cの順に接続される。これは、符号生成時と符号長を一致させてS_nを計算するためである。

【0175】そして、情報データ及び冗長データを、シンドローム/符号生成手段1802に順次入力し、前述した一復号単位の入力が終了すると、読み出したデータに対する誤りの有無を調べるため、S_n=all 0であるか否かの判定を行う（S1204）。これは、シンドローム/符号生成手段1802のシフトレジスタ部の各フリップフロップDの値で判定する。フリップフロップDの値がすべて0の場合が、S_n=all 0に相当する。

【0176】判定の結果、S_n=all 0の場合は（S1204: YES）、読み出したデータに誤りがないと判断し、内部ECC回路104における符号の情報データ部を、メモリチップ102の外部に出力し、内部ECC回路104の復号処理を終了する。

【0177】一方、S_n≠all 0でない場合は（S1204: NO）、読み出したデータに誤りが発生したと判断し、誤り位置多項式 $\sigma(x)$ を計算する。これは、図8に示した誤り評価式生成手段1804が行う。そして、誤り位置多項式 $\sigma(x)$ が求まるか否かを判別する（S1205）。

【0178】その結果、誤り位置多項式 $\sigma(x)$ が求まる場合は（S1205: YES）、誤り位置・誤り値検索手段1805を用いて、誤りの位置、値を計算し、誤り訂正処理を行う（S1206）。その後、誤り訂正処理が正常に終了したか否かを判別する（S1207）。

【0179】その結果、誤り訂正処理が正常に終了した場合は（S1207: YES）、誤り訂正の個数に応じ

25

て、行データの信頼性情報 a_i を算出する。行データ信頼性情報 a_i の算出は、行データ信頼性情報生成手段1807が行う。内部ECC回路104の訂正能力が2バイトの場合、数式(15)を用いて計算すると、 a_i の値は、訂正不能、2バイト訂正、1バイト訂正、訂正無しの順に、0、1/5、3/5、1の4値となる。

【0180】そして、内部ECC回路104における符号の情報データ部及び行データの信頼性情報 a_i を外部に出力し(S1209)、内部ECC回路104の復号処理を終了する。

【0181】一方、誤り訂正処理が正常に終了しなかった場合は(S1207:NO)、訂正能力を超える誤りが発生したと判定し、訂正不能誤り検出情報を出力し、行データの信頼性情報 a_i を0とする(S1208)。そして、内部ECC回路104における符号の情報データ部及び行データの信頼性情報 a_i を出力し(S1209)、内部ECC回路104の復号処理を終了する。

【0182】また、誤り位置多項式 $\sigma(x)$ が求まらない場合(S1205:NO)、つまり、 S_n から $\sigma(x)$ を求めた後で $\sigma(x)$ の解が求まらない場合、あるいは解が範囲外の 20 場合も、訂正能力を超える誤りが発生したと判定し、訂正不能誤り検出情報を出力し(S1208)、更に、行データの信頼性情報 a_i を0として出力する(S1209)。

【0183】以上のようにして、内部ECC回路104による復号処理が行われる。

【0184】次に、外部ECC回路1061による復号処理について説明する。

【0185】図16は、外部ECC回路1061による復号処理の流れを示す図である。

【0186】最初に、外部ECC回路1061に入力さ 30 れたデータについて、どのような復号処理を行うかを判断するため、まず、動作モード制御信号sig_1の値が、「**00」か否かを判定する(S1301)。

【0187】その結果、動作モード制御信号sig_1の値が、「**00」(具体的には、「1000」又は「0100」)の場合(S1301:YES)、外部ECC回路1061による復号処理は行われず、メモリチップ102から出力されるデータがそのままホスト等に渡される。一方、動作モード制御信号sig_1の値が「**00」ではない場合(S1301:NO)、続いて、動作モード制御信号sig_1の値が「111 40 1」か否かを判定する(S1302)。その結果、動作モード制御信号sig_1の値が「1111」の場合は(S1302:YES)、積符号復号を行う。この場合、メモリチップ102から読み出したデータを符号長42シンボルの外符号C1として復号するため、データの並べ替えを行う(S1303)。そして、並べ替えて入力したデータについて、外部ECC回路1061において積符号復号を行う(S1305)。データの並べ替え及び復号処理の詳細については、後述する。

【0188】また、動作モード制御信号sig_1の値が「1 50

26

111」でない場合は(S1302:NO)、外部ECC回路1061により、符号長(512+10)シンボルの符号(訂正能力5バイト)の復号処理を行う(S1304)。

【0189】次に、この外部ECC回路1061による5バイト訂正復号処理S1304の詳細について説明する。図17は、外部ECC回路1061による5バイト訂正復号処理の流れを示す図である。

【0190】同図に示すように、外部ECC回路1061では、まず、メモリチップ102から読み出した(512+10)シンボルのデータがシンドローム/符号生成手段1702に順次入力され、 S_n が計算される(S1401)。

【0191】そして、読み出したデータに対する誤りの有無を調べるため、 $S_n = \text{all } 0$ の判定を行う(S1402)。これは、前述した内部ECC回路104の場合と同様に、シンドローム/符号生成手段1702のシフトレジスタ部2101のフリップフロップDの値で判定する。

【0192】判定の結果、 $S_n = \text{all } 0$ である場合は(S1402:YES)、入力されたデータに誤りがないと判定し、外部ECC回路1061は、データの誤り訂正処理を行わない。

【0193】一方、 $S_n = \text{all } 0$ でない場合は(S1402:NO)、データに誤りが生じていると判定し、続けて、誤り評価式生成手段1704を用いて、誤り訂正処理を行うための誤り位置多項式 $\sigma(x)$ が求まるか否かを判定する(S1403)。

【0194】その結果、 $\sigma(x)$ が求まった場合は(S1403:YES)、誤り訂正処理を行う(S1404)。すなわち、誤り位置・誤り値検索手段1705を用いて、すべてのシンボルについて誤り検索を行った後、読み込んだデータの誤り訂正処理を行う。そして、誤り訂正処理が正常に終了したか否かを判定する(S1405)。

【0195】その結果、誤り訂正処理が正常に終了した場合は(S1405:YES)、訂正された情報データ部を出力する。

【0196】一方、誤り訂正処理が正常に終了しなかった場合は(S1405:NO)、訂正能力を超えた誤りが発生したと判定し、訂正不能誤り検出手段1707を用いて、訂正不能誤り検出情報を出力し(S1406)、情報データ部を訂正せずに出力する。

【0197】また、誤り評価式生成手段1704を用いて $\sigma(x)$ が求まらない場合も(S1403:NO)、訂正能力を超えた誤りが発生したと判定し、訂正不能誤り検出手段1707を用いて、訂正不能誤り検出情報を出力し(S1406)、復号処理を終了する。

【0198】そして、外部ECC回路1061で付加した冗長データを削除し、未訂正の情報データを出力する。

【0199】次に、外部ECC回路1061による積符

27

号の復号処理S1305の詳細について説明する。この場合、前述したように、復号処理を行う前に、メモリチップ102から読み出した512バイトの情報データと160シンボルの冗長データを外符号に変換する必要がある。

【0200】図18は、メモリチップ102から読み出された $16 \times 32 (= 512)$ シンボル(バイト)の情報データ部と $16 \times 10 (= 160)$ シンボルの冗長データ部R1からなる1次元配列データを外符号に変換する方法を示す図である。

【0201】最初に、1次元配列データから、32シンボルの外符号C1の情報データ部を抽出する。そのため、データを1シンボル読み取るごとにデータアドレスを15シンボルずつスキップする。これを32回繰り返し32シンボルの列データ(情報データ部)を生成する。

【0202】同様に、外符号C1の冗長データ部R1も、1シンボル読み取るごとにデータアドレスを15シンボルずつスキップする。これを10回繰り返し10シンボルの列データを生成する。こうして生成された10シンボルの冗長データ部と、32シンボルに変換した情報データ部とを合わせて、42シンボルの列データ(外符号)とする。

【0203】この操作をデータの先頭から順に16回繰り返す。

【0204】以上のようにして生成された外符号C1に対して、外部ECC回路1061によって復号処理を行う。図19は、外部ECC回路1061による外符号の復号処理の流れを示す図である。

【0205】同図に示すように、まず、シンδροーム／符号生成手段1702を用いて、前述した5バイト訂正復号の場合と同様に、 S_n を計算する(S1501)。そして、入力された42シンボルの符号について、誤りの有無を判別するため、 $S_n = all0$ であるか否かを調べる(S1502)。

【0206】その結果、 $S_n = all0$ の場合は(S1502: YES)、データに誤りがないと判定し、誤り訂正処理を行わず、復号処理を終了する。

【0207】一方、 $S_n = all0$ でない場合は(S1502: NO)、データに誤りが生じていると判定し、続けて、誤り評価式生成手段1704を用いて、誤り訂正処理を行うための誤り位置多項式 $\sigma(x)$ が求まるか否かを判定する(S1503)。これは、積符号の復号において、誤り位置 i の集合Fを空集合(初期値)とした場合の誤り訂正処理に相当する。

【0208】その結果、誤り位置多項式 $\sigma(x)$ が求まる場合は(S1503: YES)、次に、誤り訂正処理を行う(S1504)。すなわち、誤り位置・誤り値検索手段1705を用いて、すべてのシンボルについて誤り検索を行う。そして、数式(16)が成立するか否かを判定する(S1505)

28

5)。本実施形態の場合、 n の初期値は0で、 $d1$ は、 $11 (= 2 \times 5 + 1)$ となる。

【0209】その結果、数式(16)が成立する場合は(S1505: YES)、訂正された符号のうちの情報データ部を出力する。

【0210】一方、数式(16)が成立しない場合は(S1505: NO)、誤り位置の集合Fへ消失位置追加を行う(S1506)。ここでは、消失位置計算手段1706を用いて、集合Fに含まれない位置 i で、 a_i が0でない最小のもの(この場合、 a_i の値が $1/5$ の値をもつ情報)を誤り位置の集合Fへすべて追加する。また、 n の値をインクリメントする。

【0211】そして、消失位置 i の総数 h が $d1$ より小さいか否かを判別する(S1507)。 h が $d1$ より小さい場合は(S1507: YES)、誤り評価式生成手段1704を用いて $\sigma(x)$ を再計算し(S1503)、以後前述した処理を繰り返す。

【0212】一方、 h が $d1$ 以上の場合は(S1507: NO)、訂正能力を超えた誤りが発生したと判定し、訂正不能誤り検出手段1707を用いて、訂正不能誤り検出情報を出力し(S1508)、誤り訂正処理を行うことなく、符号から冗長データを削除し、情報データを出力した後に復号処理を終了する。

【0213】また、誤り位置多項式 $\sigma(x)$ が求まらない場合も(S1503: NO)、誤り位置の集合Fへ消失位置追加を行い(S1506)、以後前述した処理を繰り返す。

【0214】以上説明したようなフラッシュメモ리카ード501においては、メモリ103の多値記憶化に伴うデータ読み出し率エラーの増加に対して十分な誤り訂正能力を持たせる事ができる。

【0215】また、RS符号を用いた内部ECC回路104をメモリチップ102内部に設けているので、チップ単体で使用するユーザにも対応可能になる。

【0216】また、動作モード制御信号sig_1の値によって、データの符号化方法や復号方法を変更することが出来るため、メモリチップ102のデータ信頼率に合わせて、符号化率や訂正能力を変更することも可能で、記憶媒体の記憶容量の有効利用を図ることが出来る。

【0217】また、メモリチップ102の性能にばらつきがあっても、メモリチップ102内部の内部ECC回路104および外側の外部ECC回路1061の復号方法を変更することで、フラッシュメモ리카ード501全体としてのデータ信頼率を上げることが出来る。そのため、チップの歩留まり率を上げることも可能である。

【0218】なお、前述した実施形態では、積符号化時においても外符号C1の訂正能力を5バイトとしたが、積符号を用いた場合では、最大訂正能力まで訂正することで誤訂正が生じることを防ぐため、最大訂正能力まで訂正しないようにしてもよい。

【0219】また、メモリチップ102のデータ信頼率

29

によっては、積符号時は、例えば、訂正能力3バイト、冗長シンボル6シンボルとして訂正能力を変更して、符号化効率を上げることも可能である。

【0220】なお、前述したような本発明によるデジタル情報記憶装置は、図20に示すように、デジタルカメラや携帯情報端末機器、携帯電話、PHSに着脱可能な可搬型記憶装置にも使用することができる。この場合、記憶装置には、メモリチップ102のみを実装し、外部ECC回路1061で行っていた誤り訂正処理等は、PCやデジタルカメラ等の機器本体側で行うようにしてもよい。

【0221】

【発明の効果】以上詳細に説明したように、本発明では、シンボル単位訂正を行うリードソロモン符号方式を用いているので、多値技術を使った場合でも、十分なデータ信頼率が得られる。

【0222】また、本発明では、符号化方法及び復号方法を切り替えることができるので、メモリチップの記憶素子のデータ信頼率の変化に柔軟に対応することができる。

【図面の簡単な説明】

【図1】 本発明によるデジタル情報記憶装置の概要を示す図である。

【図2】 本発明によるフラッシュメモリカード501の構成を示す図である。

【図3】 フラッシュメモリカード501の具体的な実装イメージを示す図である。

【図4】 メモリチップ102の構成を示す図である。

【図5】 動作モード制御信号sig_1と各ECC回路の動作との対応関係を示す表である。

【図6】 外部ECC回路1061の構成を示す図である。

【図7】 シンドローム／符号生成手段1702の構成を示す図である。

【図8】 内部ECC回路104の構成を示す図である。

30

*【図9】 シンドローム／符号生成手段1802の構成を示す図である。

【図10】 データ書き込み時のデータパスの変化の様子を概念的に示す図である。

【図11】 データ書き込み時の符号化処理の流れを示す図である。

【図12】 実施形態で採用する積符号の構成の概念を示す図である。

【図13】 1次元配列データから外符号C1の情報データを抽出し、冗長データ部を生成する方法を示す図である。

【図14】 データ読み出し時のデータパスの変化の様子を概念的に示す図である。

【図15】 内部ECC回路104による復号処理の流れを示す図である。

【図16】 外部ECC回路1061による復号処理の流れを示す図である。

【図17】 外部ECC回路1061による5バイト訂正復号処理の流れを示す図である。

【図18】 メモリチップ102から読み出された1次元配列データを外符号に変換する方法を示す図である。

【図19】 外部ECC回路1061による復号処理の流れを示す図である。

【図20】 本発明によるデジタル情報記憶装置の適用例を示す図である。

【図21】 フラッシュメモリの構成を示す図である。

【図22】 コントロールゲート電圧とドレイン電流の関係を示す図である。

【符号の説明】

- 101 デジタル情報記憶装置
- 102 メモリチップ
- 103 メモリ
- 104 内部ECC回路
- 106、503 インターフェースLSI
- 1061 外部ECC回路

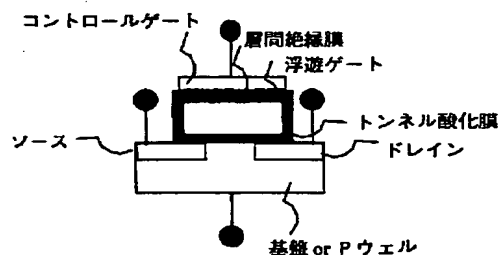
【図5】

図5

sig-1の値		ECC回路の動作(符号化時/復号時)	
上位2ビット	下位2ビット	内部ECC回路	外部ECC回路
00	10	非動作	512Byte, 5Byte 訂正
11	11	16Byte, 2Byte 訂正	82Byte, 6Byte 訂正
10	00	512Byte, 2Byte 訂正	非動作
01	00	512Byte, 6Byte 訂正	非動作
	10	復号時 2Byte 訂正 512Byte, 5Byte 訂正 復号時非動作	符号化時非動作/ 512Byte, 5Byte 訂正

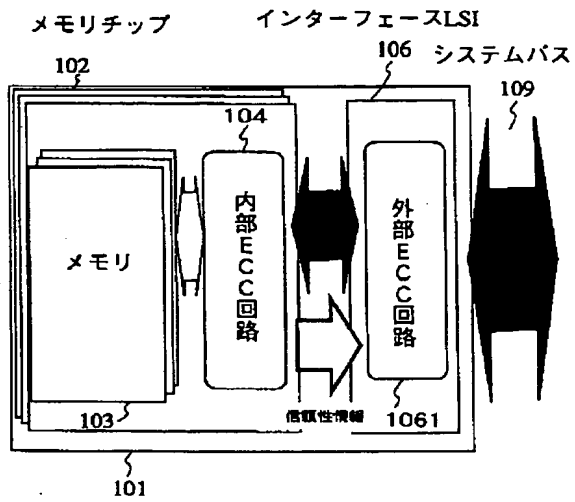
【図21】

図21



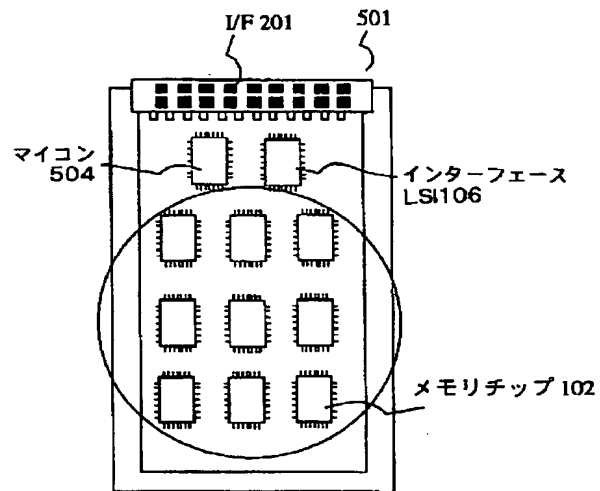
【図1】

図1



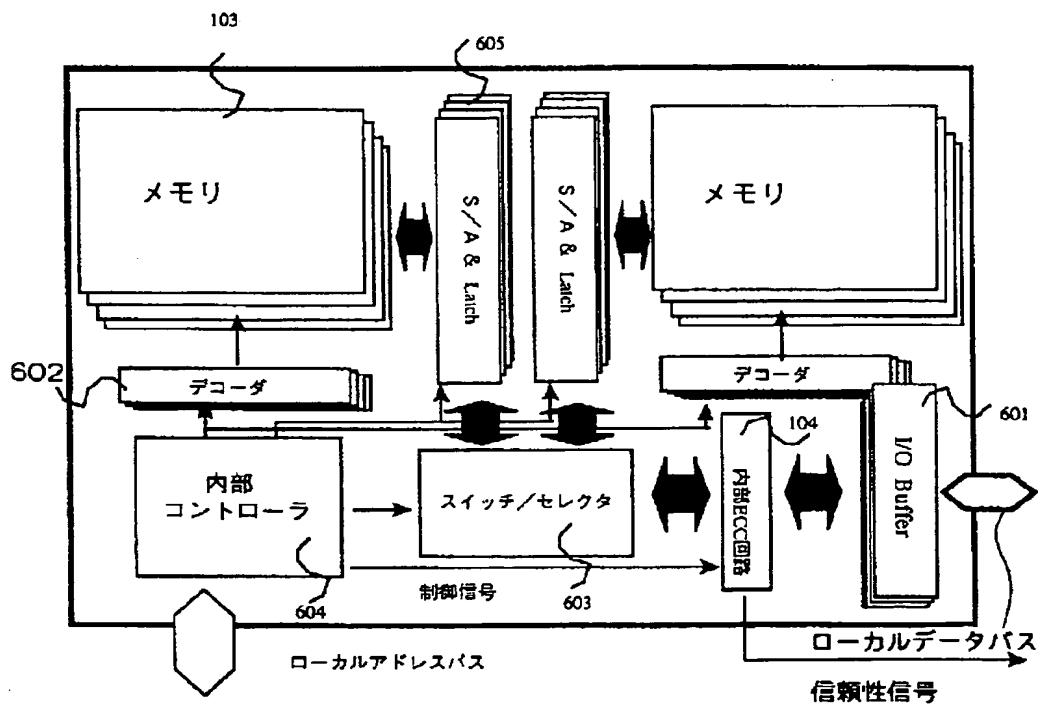
【図3】

図3



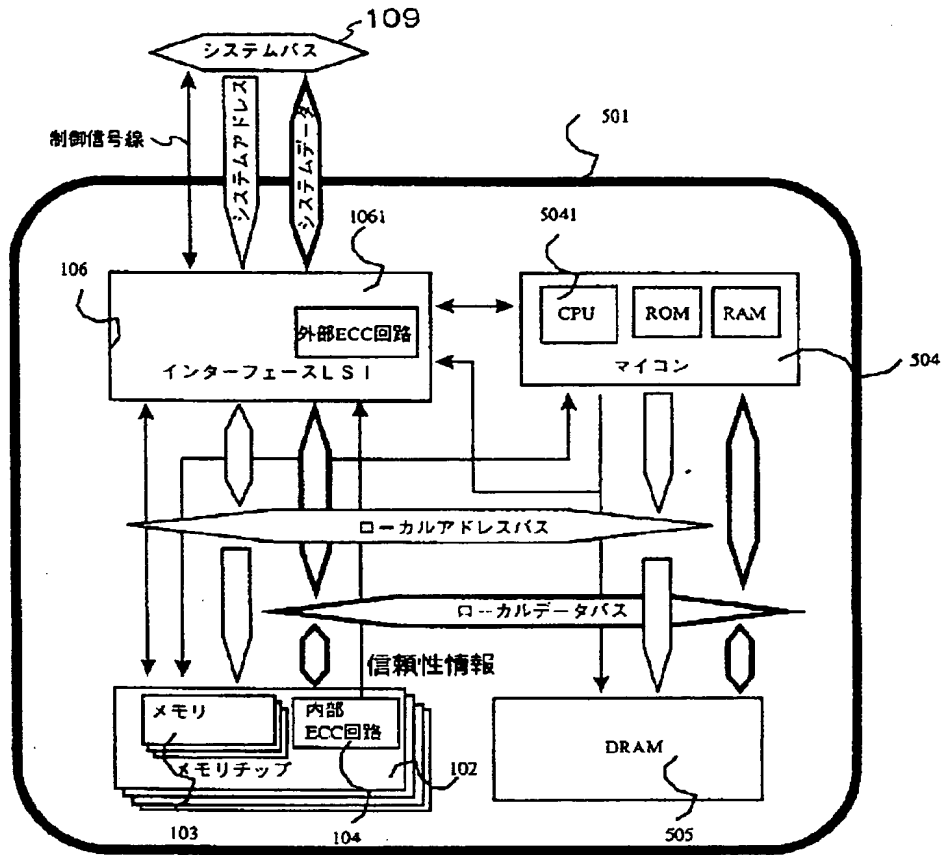
【図4】

図4

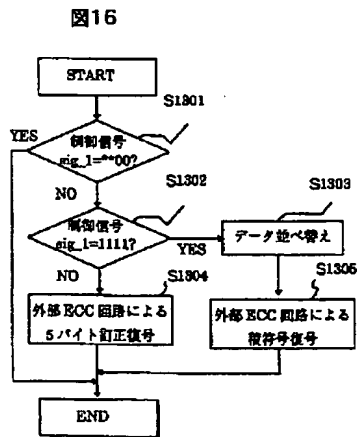


【図2】

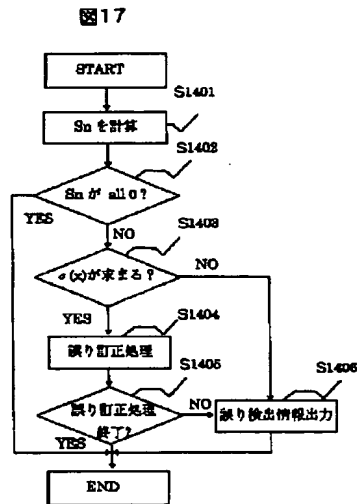
図2



【図16】

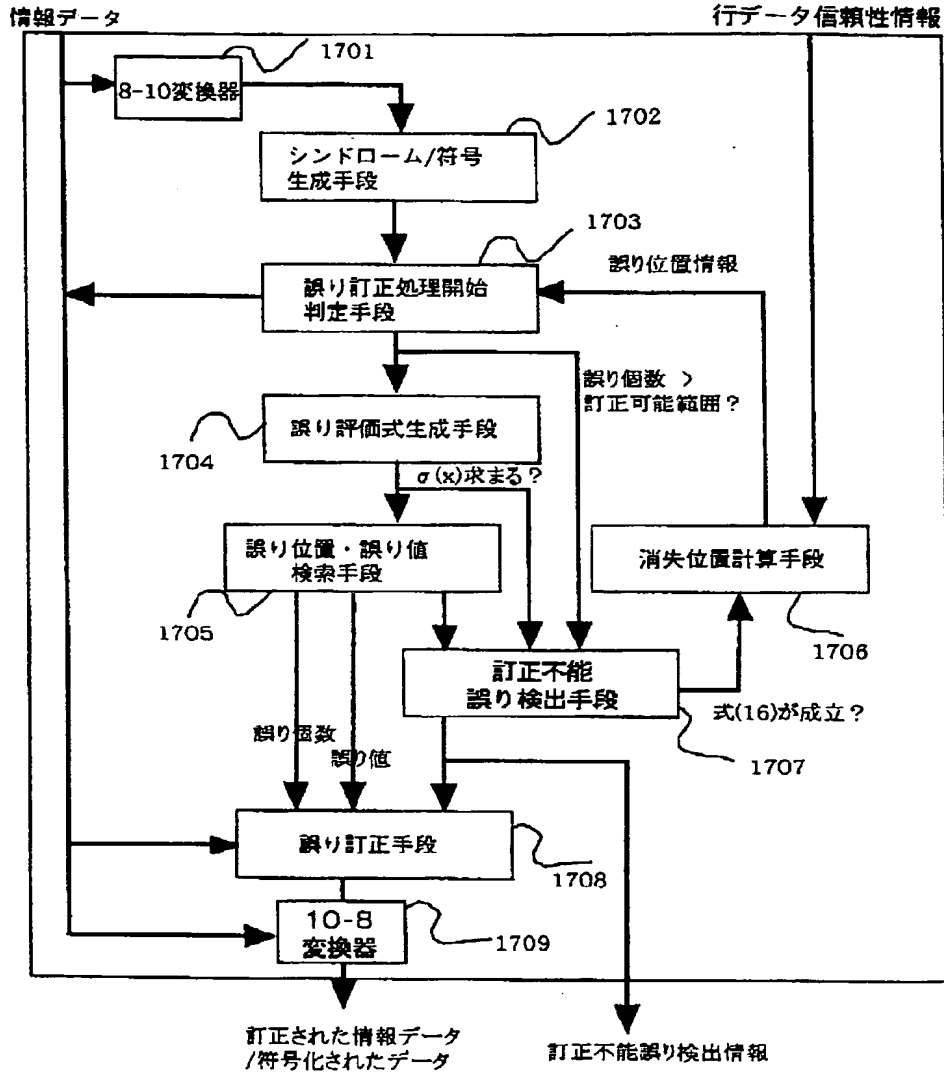


【図17】



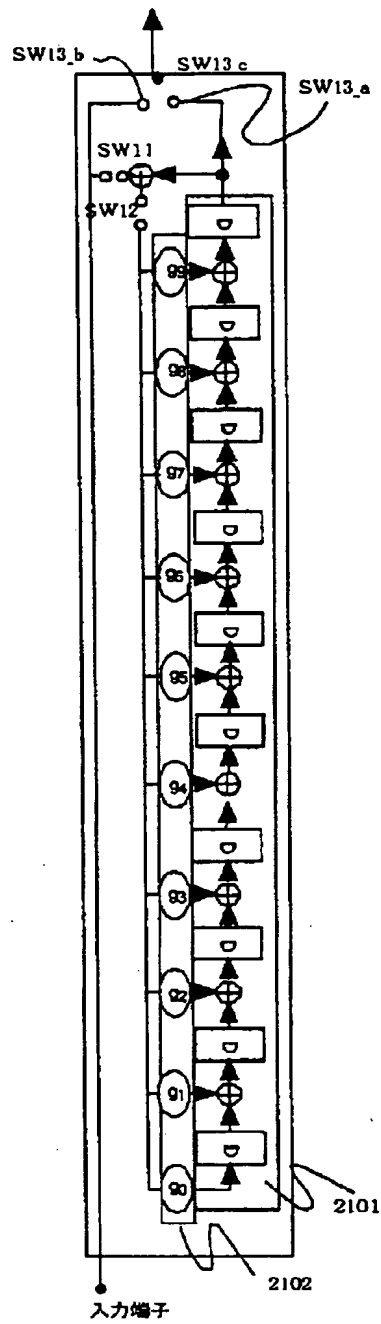
【図6】

図6



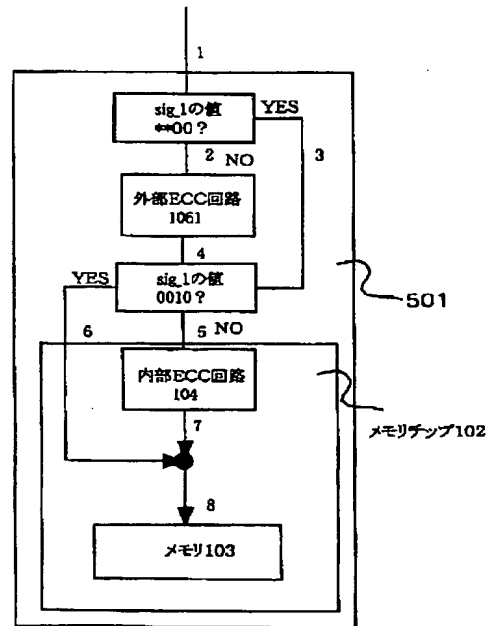
【図7】

図7



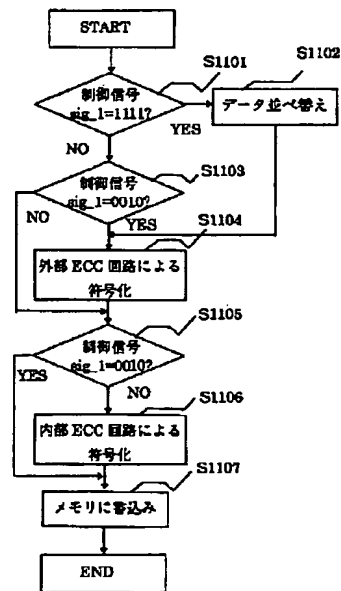
【図10】

図10



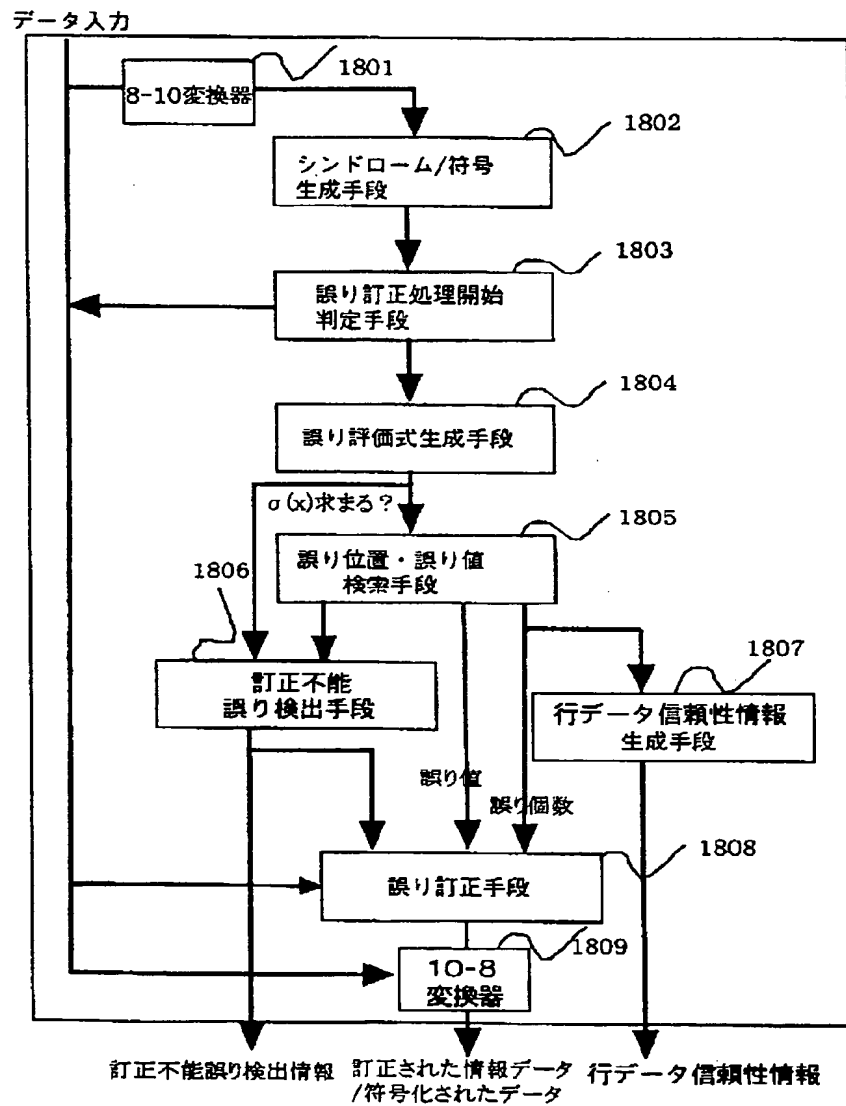
【図11】

図11



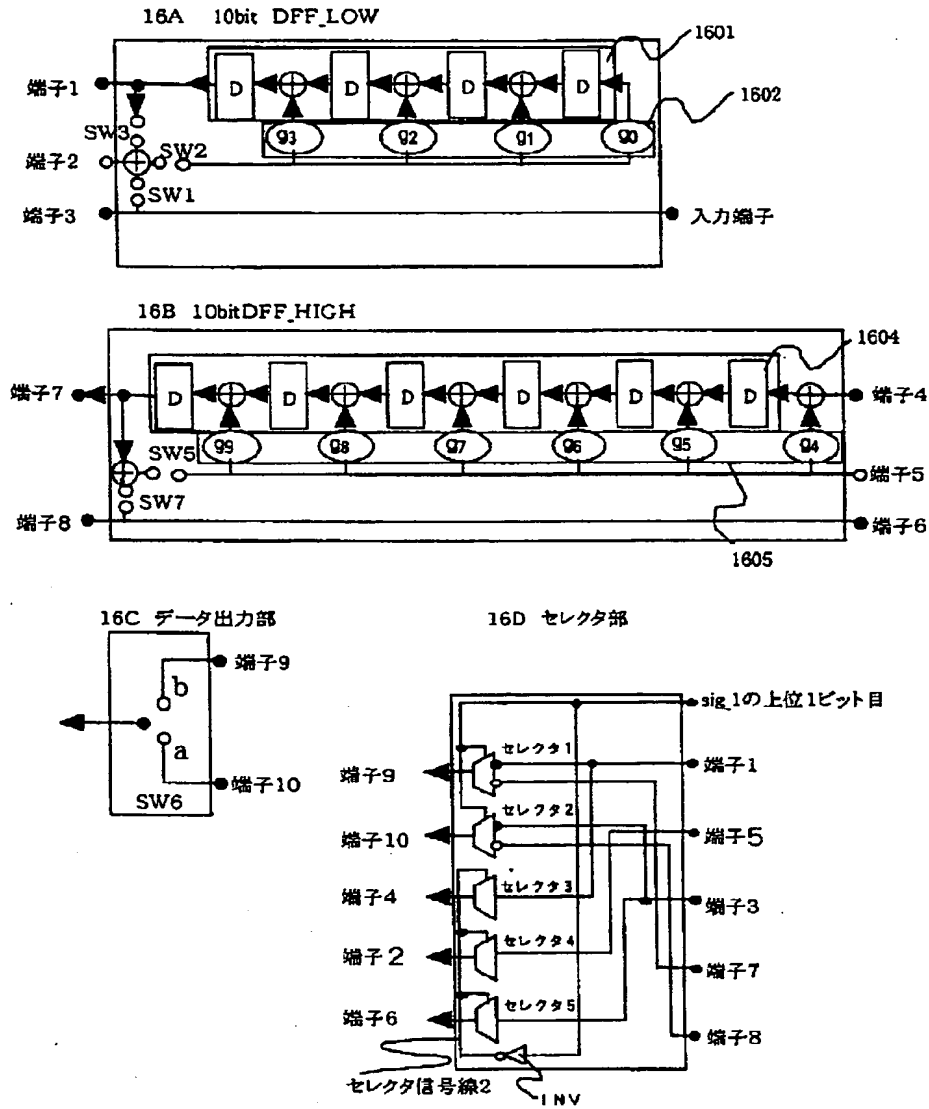
【図8】

図8

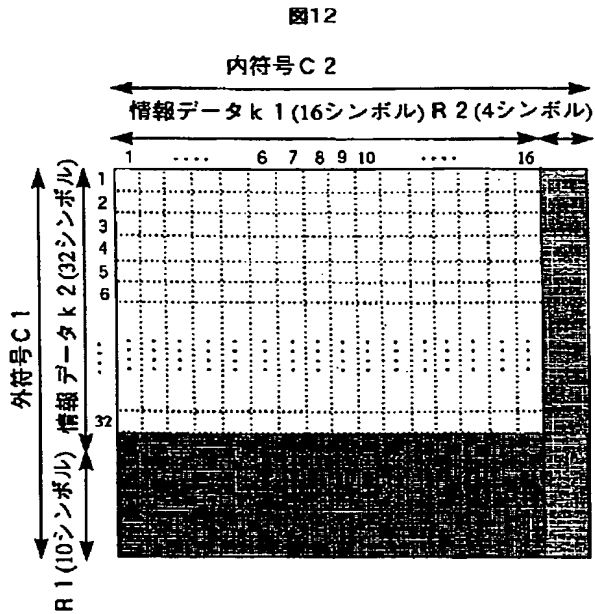


【図 9】

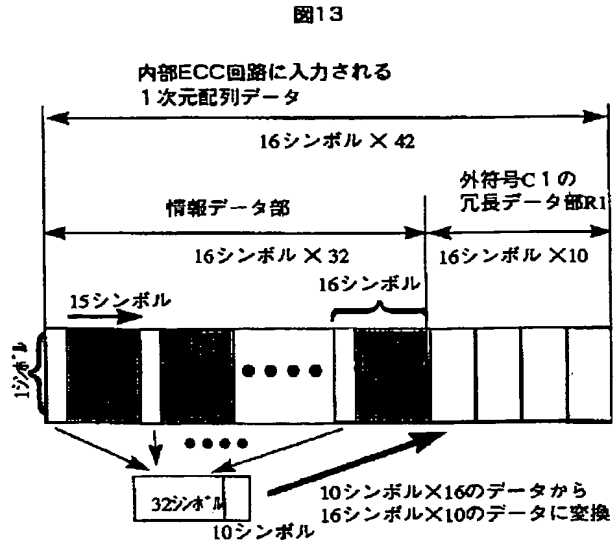
図9



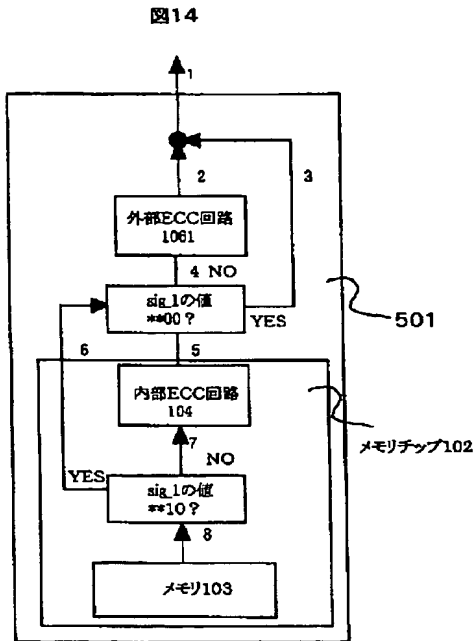
【図12】



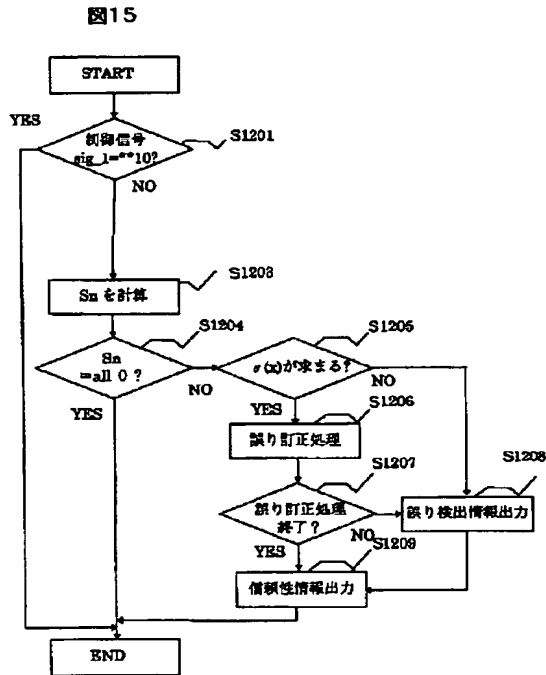
【図13】



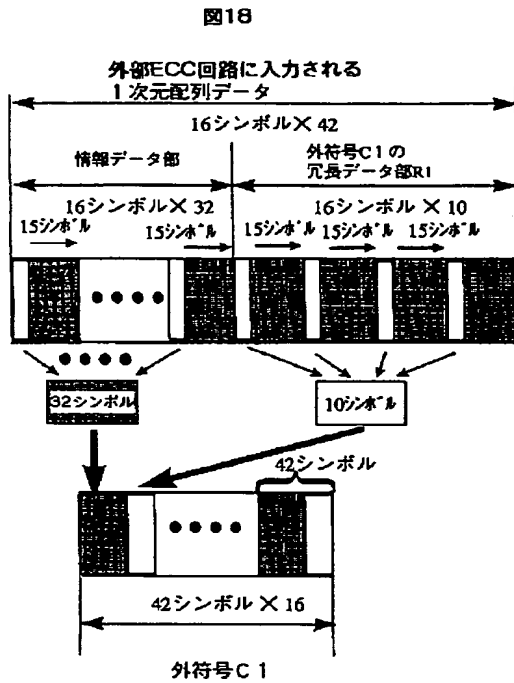
【図14】



【図15】

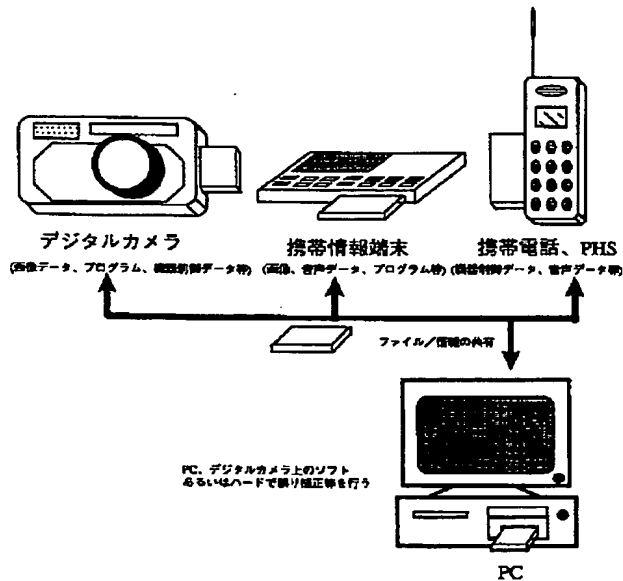


【図18】



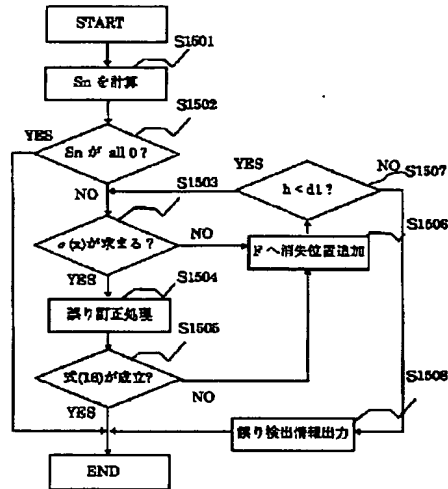
【図20】

図20



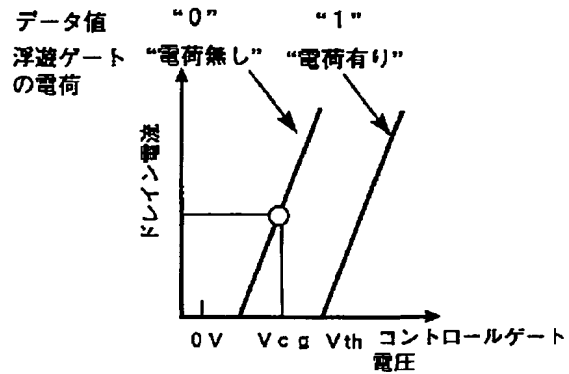
【図19】

図19



【図22】

図22



フロントページの続き

(72)発明者 辛島 哲次
東京都小平市上水本町五丁目20番1号 株式会社日立製作所半導体グループ内

(72)発明者 小谷 博昭
東京都青梅市今井2326番地 株式会社日立製作所デバイス開発センタ内

(72)発明者 野副 敦史
東京都青梅市今井2326番地 株式会社日立製作所デバイス開発センタ内

(72)発明者 西谷 卓史
神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

Fターム(参考) 5J065 AA01 AB01 AC03 AD03 AD04
AD11 AD13 AF02 AH06